A Metamodel for Annotations of Ontology Elements in OWL DL

Denny Vrandečić, Johanna Völker, Peter Haase, Thanh Tran Duc, Philipp Cimiano
{dvr,jvo,pha,dtr,pci}@aifb.uni-karlsruhe.de

Abstract: OWL DL puts several constraints on the possibilities to talk about the elements of an ontology. In particular, it is not possible to make statements about the axioms of an ontology or to make higher-order statements about the classes and properties of an ontology. This lack of expressiveness may cause problems throughout the whole lifecycle of an ontology, and especially in its practical usage. In this paper we discuss different approaches to overcome these problems. We propose a metamodel for OWL DL that allows to express statements about ontology elements, including axioms. We further describe three possible groundings of the metamodel in OWL DL and present a number of tools which we developed to support the user in working with these models. Finally, we describe some use cases for the practical application of our approach.

1 Introduction

The web ontology language OWL is the standardized ontology language for the web. It is based on both firm model theoretic semantics [HPSvH03] and existing and established web standards like XML. It allows us to define the structure and vocabulary of metadata about almost anything. We can talk about people [BM05], publication data [HEHS04], Elvis impersonators [Cha05], or almost anything else – but one thing: When talking ontologies and their elements themselves – i.e. annotating them with metadata, for example about their provenance, trustworthiness, context-dependance – we are severely restricted.

To understand these restrictions, we need to discuss the different variants of OWL. The OWL ontology language is based on a family of description logics languages. Within this family, three members have been standardized as sublanguages of OWL: OWL Full, OWL DL, and OWL Lite. These sublanguages differ in expressiveness, i.e. in their provided constructs and the allowed combinations of these constructs. OWL Full is the most expressive of the members of the OWL family. It has mainly been defined for compatibility with RDF(S) (Resource Description Framework, [BG04]). Unfortunately OWL Full is undecidable and thus impractical for applications that require complete reasoning procedures. OWL DL is a sublanguage that was designed to regain computational efficiency. OWL Lite in turn is a sublanguage of OWL DL with slightly less expressiveness.

Approaches to offer metadata about the defined classes or properties in an ontology lead the ontology to become OWL Full. This can only be avoided by using annotation properties, which constitutes non-logical information. Also, in OWL there is currently no defined way to annotate axioms, only the entities in an ontology. But in order to formalize and share information like trust, provenance, or confidence, it is not sufficient to speak only about the entities, but we must also be able to annotate the axioms of an ontology. [SM00] already argued for the need of handling axioms as objects one can refer to. This requires reification of the axioms of an ontology, which enables us to refer to these axioms as elements of the ontology.

Simple RDF reification as presented in the RDF standard [MM04] is not suitable, as it neither translates to the various OWL DL serializations, nor does it meet the goal of keeping the reification within OWL DL. Therefore in this paper we propose a form of reification compatible with OWL DL and discuss the advantages and disadvantages of our approach.

In the next section, we specify our proposed approach in terms of a metamodel, which directly extends our prior work on a metamodel for OWL DL based on the Meta Object Facility (MOF). In Section 3 we propose several possible groundings of this metamodel, which are either directly compatible with OWL DL or require slight extensions to the language. We describe the possible use cases of the given approach in Section 4. Before we summarize the conclusions of this paper, we compare our approach to related work.

2 A Metamodel of OWL DL Ontologies

In this section we introduce a metamodel for OWL DL that addresses the requirements for annotating ontology elements identified in the previous sections. We rely on the metamodeling features of Model Driven Architecture (MDA) [MSUW02], which provide the means for the specification of modeling languages in a standardized, platform independent manner. The metamodel extends our previous work on metamodels for OWL reported in [BVEL04] and [BH06], which we refer to for a complete reference. In this paper, we show a small part of the metamodel focusing on the extensions relevant for annotations of ontology elements. The rest of this section will provide a summary of the OWL language whilst introducing our metamodel. Interested readers may refer to the specifications [DS04] for a full account of OWL.

Ontologies URIs are used to identify all objects in OWL. Figure 1 shows the central part of the OWL DL metamodel. Among others, it shows that every element of an ontology is a subclass of the class `OntologyElement` and hence a member of an `Ontology`. In particular, also the class `Axiom` is a subclass of `OntologyElement` and as such first-class objects in the metamodel.

Axioms Axioms describe relations between other ontology elements, be it properties, classes or individuals. Basically, an ontology may be regarded as a (named) set of axioms. OWL provides a number of class axioms, property axioms, and axioms to state facts about individuals. For a complete reference we refer the reader to [DS04]. In our metamodel we consider all these axioms as `OntologyElement` and thus as objects that can be directly

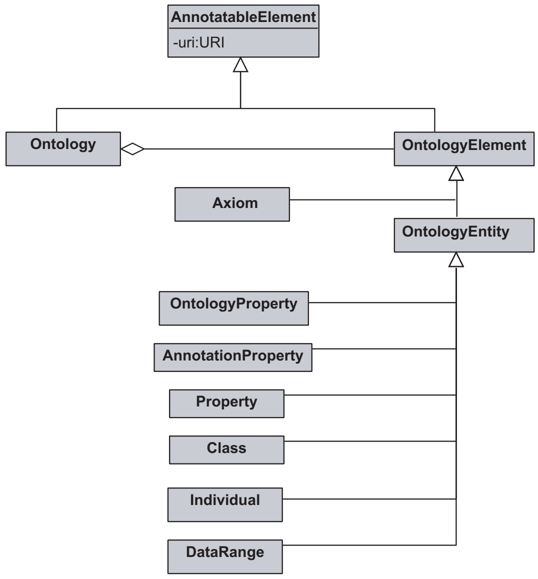

Figure 1: Main Elements of the Ontology Definition Metamodel

referenced using a URI.

Properties Properties represent named binary associations in the modeled knowledge domain. OWL distinguishes two kinds of properties, so called object properties and datatype properties. Both are generalized by the abstract metaclass `Property`. Properties can be functional, i.e. their range may contain at most one element. Their domain is always a class. Object properties may additionally be inverse functional, transitive, symmetric or inverse to another property. Their range is a class, while the range of datatype properties is a datarange.

Users can relate properties by using two axioms: Property subsumption (`subPropertyOf`) specifies that the extension of a property is a subset of the related property. Similarly, property equivalence (`equivalentProperty`) defines extensional equivalence. OWL DL disallows that object and datatype properties are related via axioms.

Annotation properties Given elements of an OWL ontology can be annotated with metadata. Several annotation properties, e.g. `owl:versionInfo`, are predefined and users can define further annotation properties. We treat annotation properties similarly to ontology properties. However, the subject of an `AnnotationPropertyValue` is an `AnnotateableElement` and the object is a Annotation, which can be either a `DataValue`, a `URI` or an `Individual` (cf. Figure 2).

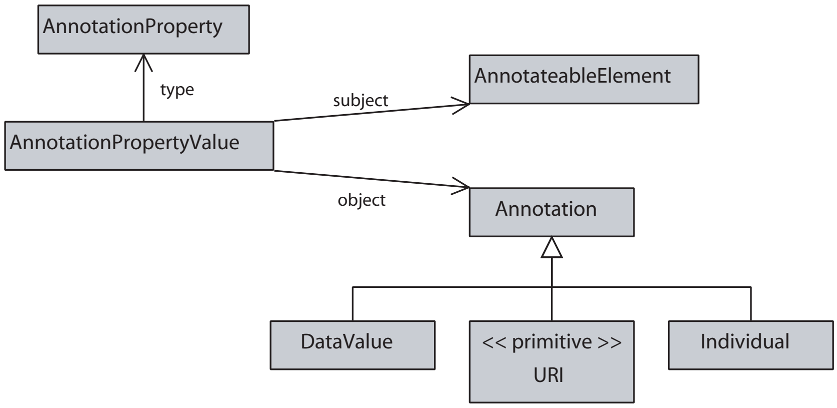

Figure 2: Annotations

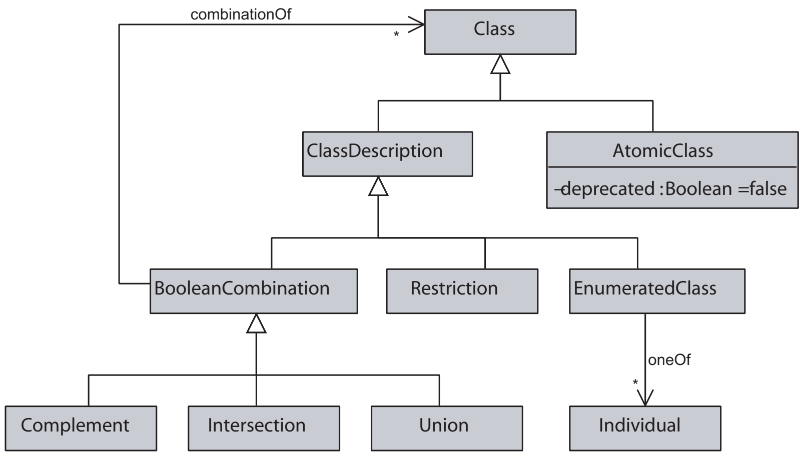

Figure 3: Class constructors

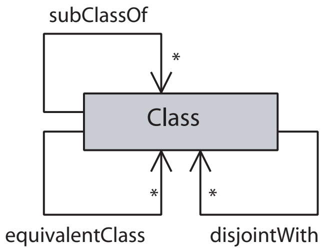

Figure 4: Class axioms

Class Constructors In OWL DL classes can be formed with several class constructors (cf. Figure 3). One can conceptually distinguish the boolean combination of classes, restrictions and enumerated classes. `EnumeratedClass` is only available in OWL DL and is defined through a direct enumeration of named individuals. Boolean combinations of classes are provided through `Complement`, `Intersection` and `Union`.

Figure 4 shows that classes can be related with each other using class axioms, such as class subsumption (`subClassOf`), class equivalence (`equivalentClass`). These relations between classes are naturally modeled as associations.

3 Grounding

In this section we propose a number of different possible groundings of the metamodel in the OWL DL language. The first approach relies on a meta-ontology to ground the metamodel in a way compatible with the current OWL 1.0 standard. The other two groundings require an extension of OWL 1.0 by introducing URIs for axioms in the ontology.

3.1 Annotation Compatible with OWL 1.0

We have defined an OWL DL meta-ontology to capture the metamodel as presented in the previous section. In this ontology we explicitly also capture axioms. Ontologies can be transformed to become instance data with regards to the vocabulary of the meta-ontology. We here first present a small fragment of the meta-ontology that reflects the part of the metamodel shown in Figure 4. Then we will give a small example, how an ontology looks like when transformed. Note that axioms, in this case, translate to proper individuals as well, and thus become annotatable, as we will show in the example.

$(1)\textsf{Class} \sqsubseteq \textsf{OntologyEntity}$
$(2)\textsf{Axiom} \sqsubseteq \textsf{OntologyElement}$
$(3)\textsf{SubClassOfAxiom} \sqsubseteq \textsf{Axiom}$
$(4)\textsf{EquivalentClassAxiom} \sqsubseteq \textsf{Axiom}$
$(5)\textsf{DisjointWithAxiom} \sqsubseteq \textsf{Axiom}$

$(6)\top \sqsubseteq \forall \textsf{subClassOfSubClass}.\textsf{Class}$
$(7)\top \sqsubseteq \forall \textsf{subClassOfSubClass}^{-1}.\textsf{SubClassOfAxiom}$
$(8)\top \sqsubseteq \leq 1\ \textsf{subClassOfSubClass}.\top$
$(9)\top \sqsubseteq \forall \textsf{subClassOfSuperClass}.\textsf{Class}$
$(10)\top \sqsubseteq \forall \textsf{subClassOfSuperClass}^{-1}.\textsf{SubClassOfAxiom}$
$(11)\top \sqsubseteq \leq 1\ \textsf{subClassOfSuperClass}.\top$
$(12)\top \sqsubseteq \forall \textsf{equivalentClass}.\textsf{Class}$
$(13)\top \sqsubseteq \forall \textsf{equivalentClass}^{-1}.\textsf{EquivalentClassAxiom}$
$(14)\top \sqsubseteq \forall \textsf{disjointWith}.\textsf{Class}$
$(15)\top \sqsubseteq \forall \textsf{disjointWith}^{-1}.\textsf{DisjointWithAxiom}$

Axioms 1-5 define the terms used. Every axiom type is defined by a class of its own (refer to the following example). The rest of the axioms defines the domain and ranges of the used properties. The following is an example of a very simple ontology, with just one axiom that states that all persons are living beings.

$\textsf{Person} \sqsubseteq \textsf{Mortal}$

Using the meta-ontology, we can represent this ontology as follows:

$\textsf{Class}(\textit{Mortal})$
$\textsf{Class}(\textit{Person})$
$\textsf{SubClassOfAxiom}(\textit{axiom1})$
$\textsf{subClassOfSuperClass}(\textit{axiom1}, \textit{Mortal})$
$\textsf{subClassOfSubClass}(\textit{axiom1}, \textit{Person})$
$\textsf{subClassOf}(\textit{Person}, \textit{Mortal})$

Note that the class PERSON is something else than its representing individual in the meta-ontology, which is the individual *Person*. The axiom of the original ontology is reified explicitly as the individual *axiom1*, an instance of the class SUBCLASSOFAXIOM (as it is an axiom that represents a subclass relation between to classes). The axiom is connected to the entities taking part in that axiom with the given properties (i.e., the SUBCLASSOF-SUPERCLASS property points to the superclass in the axiom, and the SUBCLASSOF-SUBCLASS to the subclass, respectively). The last axiom offers a direct property instance representing the original axiom. This is required for some modeling tasks, as will be shown in the use case in Section 4.3.

Now it is possible to state further facts about this axiom, like its source or the confidence we put into the axiom, within the ontology:

$\textsf{creator}(\textit{axiom1}, \textit{Aristotle})$
$\textsf{confidence}(\textit{axiom1}, 0.95)$

Naturally, we also can talk about the entities of the ontology in the same manner:

$\textsf{creator}(\textit{Person}, \textit{God})$

We have implemented the presented approach of creating a meta-ontology out of OWL DL ontologies within OWL DL. The implementation is available as part of the KAON2

OWL tools[1], and is based on the KAON2 reasoner and ontology management infrastructure [HMS04]. Two tools are currently implemented:

- `owl meta` creates a meta-ontology from an input ontology. As described in Section 2, both the ontological entities like individuals, properties and classes, as well as the axioms of the ontology are transformed. The ontology is transformed in such a way that it can be safely merged with the original ontology itself, thus allowing for rich metadata within the ontology without breaking it.
- `owl demeta` on the other hand takes a meta-ontology and creates the ontology that would have caused the given meta-ontology.

A problem with the meta-ontology is that it is considerably bigger than the original ontology. As every ontology entity is described with an extra axiom, and every already existing axiom is described with at least two further axioms, the meta-ontology often is three or four times as big as the original ontology. In the given use cases (Section 4), the meta-ontology is only processed automatically, and not presented to the user. Therefore, it remains a question of scalability of the tools and higher requirements regarding resources, but the user does not have to deal with the growing size of the ontology. The actual growth rate depends on structural properties of the ontology, but is basically a constant factor to the size.

3.2 Annotations of Axioms with URIs

In this section, we discuss two further possible groundings for the proposed metamodel. In contrast to the one above, these approaches require an extension of the current OWL 1.0 standard in order to assign URIs to axioms. That is, axioms would become entities in the ontology that can be referred to, i.e. they are reified in the ontology. In the following, we denote the URI of an axiom within square brackets following the axiom[2]:

$\textsf{Person} \sqsubseteq \textsf{Mortal}$ [*axiom1*]

We are now able to refer to the axiom using its URI. Depending on how this is done, we distinguish two approaches discussed in the following.

3.2.1 Annotations using a Meta-Ontology

In this approach, a separate meta-ontology in OWL DL is employed in order to make statements about ontology elements in a similar way as in the approach described in Section 3.1. However, it is not required anymore to reproduce the entire ontology within the meta-ontology, as we are now able to refer to all elements of the original ontology – including its axioms – via their URI:

[1] `http://owltools.ontoware.org`

[2] For the serialization of this extension in an XML based syntax, we suggest the approach taken in the current version of the syntax of SWRL [HPSB+04], a rule extension of the OWL language: SWRL allows such URI references as an optional element, which can be used to identify the corresponding rule.

$\text{AXIOM}(\textit{axiom1})$
$\text{CREATOR}(\textit{axiom1}, \textit{Aristotle})$
$\text{CONFIDENCE}(\textit{axiom1}, 0.95)$

Please note that the metadata about the ontology elements constitutes logical information. This also implies that the metadata should be kept in a separate ontology in order to avoid semantic conflicts. Such conflicts can easily occur when axioms of different metalevels are combined, as shown in the following example:

$\top \equiv \{\textit{Adam}, \textit{Eva}\}\ [\textit{axiom2}]$
$\text{AXIOM}(\textit{axiom2})$

The first axiom enumerates the individuals of the domain under consideration and assumes a fixed number of objects in the universe of discourse. However, the second axiom reifies the first axiom as an individual in the same universe of discourse. As a consequence, we would derive that the *axiom2* is the same individual as either *Adam* or *Eva*, which is clearly undesired.

3.2.2 Annotations using Annotation Properties

The final grounding we propose abandons the use of a meta-ontology. Instead, annotation properties are used for the representation of metadata about ontology elements. In the current version of the specification of OWL DL, it is possible to annotate ontology entities (i.e., classes, properties and individuals) in this way. We here propose to allow such annotations also for axioms. The above example would then look as follows:

$\text{PERSON} \sqsubseteq \text{MORTAL}\ [\textit{axiom1}]$
$\texttt{CREATOR}(\textit{axiom1}, \textit{Aristotle})$
$\texttt{CONFIDENCE}(\textit{axiom1}, 0.95)$

In contrast to the grounding described in Section 3.2.1, CREATOR and CONFIDENCE here denote *annotation properties*. These annotations constitute non-logical information, meaning that their treatment is outside the (regular) semantics of OWL. They can thus be stored within the original ontology without causing semantic conflicts.

4 Use Cases

There are several scenarios for the usage of modeling metadata about ontology elements. We will present some of them in detail in this section and discuss several more ideas.

4.1 Modeling Uncertainty in Ontology Learning

Ontology learning frameworks such as OntoLT [BOS03], OntoLearn [NVCN04], ASIUM [FN98] or Text2Onto [CV05] aim at the semi- or even fully automatic extraction of ontologies from sources of textual data. Machine learning and natural language processing techniques are used to identify classes, instances, as well as various types of taxonomic (`subclass-of`, `instance-of`) and non-taxonomic (e.g. `part-of`) relationships in a corpus of unstructured or semi-structured documents. But the ontologies which are generated by these systems often lack precise information about the source and the reliability of the individual ontology elements (entities and axioms). This kind of information is essential, if the user wants to revise the ontology after the ontology learning process. Therefore, Text2Onto associates with each of the learned ontology elements a confidence and a relevance value which allow for a more target-oriented inspection of the learned ontology. Both confidence and relevance values can be considered as some sort of rating annotations defined as follows:

Let N denote the set of all possible ontology elements, then an *ontology rating annotation* is a partial function $r : N \to \mathbb{R}$.

- We use a **confidence** rating r_{conf} to indicate how confident the system is about the correctness of an ontology element. The confidence rating is determined by the number and quality of evidences found in the corpus.
- We use a **relevance** rating r_{rel} to denote the relevance of an ontology element with respect to a particular domain given by the corpus.

In addition to confidence and relevance values each ontology element is associated with evidences and references that can be used to generate formal or informal explanations for particular results.

Evidences are the basis for any computation of a confidence value. Typical examples for evidences which may lead to the creation of a `subclass-of` relation, for instance, include Hearst patterns [Hea98] and hyponymy relationships in WordNet [Fel98].

References are pointers to occurrences of the regarding element in the corpus or other underlying knowledge sources such as ontologies or lexical resources. Adding a list of references to each ontology element not only increases the traceability of the ontology learning process, but also facilitates the detection of ontology changes in case of changes to the corpus.

Further extensions to the metamodel have been made to represent information about the algorithms which suggested the addition or removal of a particular ontology element, learning parameters and timestamps.

4.2 Modeling Provenance for Knowledge Fusion

In many cases, the information derived from diverse sources leads to inconsistencies. This is especially the case if information is derived using automatic knowledge acquisition tools such as wrappers [FK00] or information extraction systems (e.g. [Cir01], [BCRS06]). An important problem is thus to deal with and resolve inconsistencies arising from the integration of information extracted from different sources as well as by different agents (human or machines). Knowledge fusion is in fact one of the central topics of the X-Media[3] project. Fusing knowledge extracted from heterogeneous sources in essence requires (i) an algorithm to pinpoint down where the inconsistencies arise, (ii) a procedure to resolve inconsistencies by removing axioms leading to these inconsistencies, as well as (iii) a representation of the provenance of axioms on the basis of which to decide which axioms should be removed.

In the X-Media project, we thus have the requirement that axioms need to be annotated with provenance information. Currently, we annotate an axiom as follows:

$\textsc{hasProvenance}(\textit{axiom1}, \textit{p1})$
$\textsc{hasSource}(\textit{p1}, \textit{s1})$
$\textsc{hasCertainty}(\textit{p1}, \textit{c1})$
$\textsc{hasTimeStamp}(\textit{p1}, \textit{t1})$

This information can then guide the inconsistency resolution procedure in the decision which axioms should be removed first. For our knowledge fusion scenario, we intend to build on the approach of [HV05] to find *minimally inconsistent axioms sets* within a given ontology. The idea behind minimally inconsistent axiom sets is that the removal of one axiom in each set will lead to consistency. The decision which axiom to remove can then indeed be guided by the provenance information as described above. The annotation of axioms with provenance information can be certainly done using annotation properties (see above). However, it could be also interesting to internalize the inconsistency resolution process into the meta-ontology.

In general, the reification of axioms would allow to easily pose queries asking for all the axioms with a minimum certainty, all the axioms derived by a certain agent or a certain source, all the axioms with a time-stamp after a certain date etc.

4.3 Reasoning with Metaproperties for Ontology Evaluation

OntoClean [GW00] is a well-known methodology for the formal analysis of taxonomic relationships. The evaluation of an ontology by means of OntoClean requires two steps: First, every class has to be tagged with respect to a predefined set of metaproperties (i.e. does the class belong to a specific metaclass or not) including *rigidity*, *unity* and *identity*. Second, all subsumption (i.e. `subclass-of`) relationships are checked against a number of constraints disallowing certain combinations of tagged classes.

For example, let us consider the metaproperty of **rigidity** (*R*) which is based on the notion of *essence*. A concept is essential for an instance *iff* it is necessarily an instance of this concept, in all worlds and at all times. *Iff* a concept is essential to all of its instances, the concept is called **rigid** and is tagged with *+R*. *Iff* it is not essential to some instances, it is called **non-rigid**, tagged with *-R*. An **anti-rigid** concept is one that is not essential to all of its instances. It is tagged $\sim$*R*. An example of an anti-rigid concept would be *teacher*, as no teacher has always been, nor is necessarily, a teacher, whereas *human* is a

[3] http://nlp.shef.ac.uk/X-Media/

rigid concept because all humans are necessarily humans and neither became nor can stop being a human at some time.

Based on the notion of rigidity OntoClean defines the following constraint which must hold for all taxonomic relationships in the ontology: **∼*R* can't subsume +*R*.** This means that having a concept *C* subsuming the concept *D*, with *C* tagged ∼*R* and *D* tagged +*R*, would lead to the following inconsistency: *D* must always hold true for all of its instances. *D*, as a subsumed concept, would always imply *C* for all of its instances. Therefore there are at least some instances of *C* that are necessarily *C* as they are *D*. Thus *C* can not be anti-rigid, as the tagging says, because this would mean that it is not necessarily true for any of its instances – which would be a contradiction. The classic example is *student*, an anti-rigid concept, subsuming *human*, a rigid concept, which is obviously wrong: whereas every student is free to leave the university and stop being a student, humans cannot stop being humans. As every human would be a student, according to the example, they never could stop being a student, which contradicts the previous sentence.

This example shows that the notion of formal consistency as defined by the OntoClean methodology is not only of philosophical or theoretical relevance, but closely relates to the logical consistency of ontologies, thus being of major importance for all reasoning-based applications. Obviously, tagging all classes of an ontology with appropriate Onto-Clean metaproperties is difficult and time consuming. Therefore recent approaches such as AEON [VVS05] try to automate the tagging process.

Tagging the ontology is only the first (and most expensive) step in evaluating an ontology with OntoClean. The second is to check if the taggings are allowed by the OntoClean rules. This can – and has been done – automatically, for example [FLGP02] describes such an approach. But using the meta-ontology approach described in this paper, we can actually check the OntoClean rules within OWL DL. First we state the following additional axioms:[4]

$\text{RigidClass} \sqsubseteq \text{Class}$
$\text{NonrigidClass} \sqsubseteq \text{Class}$
$\text{RigidClass} \sqcap \text{NonrigidClass} \sqsubseteq \bot$
$\text{RigidClass} \sqsubseteq \forall \text{subClassOf}.\text{RigidClass}$

Now we can create a meta-ontology of the ontology to be evaluated, including the appropriate taggings.

$\text{RigidClass}(Person)$
$\text{NonrigidClass}(Student)$
$\text{subClassOf}(Student, Person)$

Combining this meta-ontology with the ontology formalizing the OntoClean rules, a reasoner will discover that the ontology is not satisfiable, i.e. that the ontology does not satisfy the OntoClean rules. The first advantage is that we can find this out without any further implementation effort, but just using standardized OWL DL reasoners. The second advantage is that we transformed the problem of not meeting the OntoClean meta rules

[4]Note that for the case of clarity we use a slightly simplified OntoClean rule, ignoring the idea of anti-rigidity as described in [GW04]

into a logical unsatisfiability. This also means that we can now use standard OWL debugging tools and techniques (like those presented in [LSV04, KPSH05, Sch05]) to resolve the problems that would have required a dedicated set of tools otherwise.

5 Related Work

Metamodeling as a feature of knowledge representation can be found in a number of related languages and approaches. OWL Full, which inherits the full modeling features of RDF(S) including the features of reification and the possibility to freely mix modeling layers, trivially allows to express the metamodeling features of our approach. As argued before, the undecidability of OWL Full and the missing reasoning support make this OWL Full impractical for our purposes. In [PH02] the authors recognize the need for a clean metamodeling architecture for ontology languages and propose such an architecture with fixed metamodeling for RDF(S), called RDFS(FA). [PHS05] proposes OWL FA, an extension of OWL DL with the metamodeling architecture of RDFS(FA). [Mot05] analyzed the properties of metamodeling within OWL and proposed two alternative semantics, which allow to retain decidability. However, none of the latter approaches allows to state facts about axioms; further, there is currently no tool support.

Several other works address the annotation of complete ontologies, and offering vocabularies and models of ontologies. But these aim at a much coarser level and do neither regard single ontological entities nor axioms. Work in this direction includes the Ontology Metadata Vocabulary OMV [HSH^{+}05] or the O^2 ontology metamodel [GCCL06].

There also exists a line of related work relying on the metamodeling features of MOF, as pursued in our approach. These include the Ontology Definition Metamodel (ODM) of the OMG[5] and the metamodel proposed in [BVEL04] and [BH06], which is extended by our approach. To our knowledge, our metamodel is the first one to equip the language with the ability to state facts about any ontology elements, including axioms, and to provide a language compatible with OWL DL.

6 Conclusion

We have presented an approach to annotate ontology elements – in particular including axioms – in OWL DL ontologies. This allows us to state rich metadata about elements in OWL DL ontologies. We have shown how this enables typical use cases in the Semantic Web.

We think that the approach presented here has several merits. As we have seen in the use cases, (Section 4.3) we can take advantage of more expressive constructs than the simple annotations already allowed by OWL DL.

Some of the use cases presented require the ability to annotate axioms. Because the

[5] `http://www.omg.org/ontology/`

grounding compatible with the current OWL 1.0 standard is very verbose and cumbersome, we think that future OWL versions should be extended in such a way to allow giving axioms a URI. Instead of generating new URIs for the axioms in the meta-ontology, we can simply reuse the URIs provided by the ontology itself. This will foster reuse, exchange and integration of rich metadata about ontology elements.

The ontology metamodel and the supporting implementation supports several use cases. We have presented a few possible use cases, but expect many further to emerge. The initial building, analysis and maintenance of ontologies can be supported and made explicit by the usage of our approach. It enables the practitioner to take advantage of the full expressivity of OWL DL and to reuse already developed tools in a novel way.

Acknowledgements Research presented in this paper was supported by the European Commission under contract IST-2003-506826 SEKT (http://www.sekt-project.org/), IST-FP6-026978 X-Media (http://www.x-media-project.org/), and IST-2006-027595 NeOn (http://www.neon-project.org/). The expressed content is the view of the authors but not necessarily the view of the projects as a whole. We wish to thank our colleagues for extensive discussions and the reviewers for valuable comments.

References

[BCRS06] P. Buitelaar, P. Cimiano, S. Racioppa, and M. Siegel. Ontology-based information extraction with SOBA. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2006.

[BG04] D. Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004. Available at http://www.w3.org/TR/rdf-schema/.

[BH06] Saartje Brockmans and Peter Haase. A Metamodel and UML Profile for Rule-extended OWL DL Ontologies –A Complete Reference. Technical report, Universität Karlsruhe, March 2006. `http://www.aifb.uni-karlsruhe.de/WBS/sbr/publications/owl-metamodeling.pdf`.

[BM05] Dan Brickley and Libby Miller. The Friend Of A Friend (FOAF) Vocabulary Specification, July 2005. Namespace Document 27 July 2005 ('Pages about Things' Edition).

[BOS03] P. Buitelaar, D. Olejnik, and M. Sintek. OntoLT: A Protégé Plug-In for Ontology Extraction from Text. In *Proc. of the 2nd Int. Semantic Web Conference (ISWC2003)*, 2003.

[BVEL04] S. Brockmans, R. Volz, A. Eberhart, and P. Loeffler. Visual Modeling of OWL DL Ontologies using UML. In F. van Harmelen, S. A. McIlraith, and D. Plexousakis, editors, *The Semantic Web – ISWC 2004*, pages 198–213. Springer-Verlag, 2004.

[Cha05] Bob Du Charme. Metadata about Elvis impersonators, March 2005. available at http://www.snee.com/rdf/elvisimp.rdf.

[Cir01] F. Ciravegna. Adaptive Information Extraction from Text by Rule Induction and Generalization. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1251–1256, 2001.

[CV05] Philipp Cimiano and Johanna Völker. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. In Andres Montoyo, Rafael Munoz, and Elisabeth Metais, editors, *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238, Alicante, Spain, JUN 2005. Springer.

[DS04] M. Dean and G. Schreiber, editors. *OWL Web Ontology Language Reference*. 2004. W3C Recommendation 10 February 2004.

[Fel98] C. Fellbaum. *WordNet, an electronic lexical database*. MIT Press, 1998.

[FK00] F. Freitag and N. Kushmerick. Boosted Wrapper Induction. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*, pages 577–583, 2000.

[FLGP02] M. Fernández-López and A. Gómez-Pérez. The integration of OntoClean in WebODE. In *Proc. of the EON2002 Workshop at 13th EKAW*, 2002.

[FN98] D. Faure and C. Nedellec. A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology. In *Proceedings of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications*, 1998.

[GCCL06] Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann. Qood grid: A metaontology-based framework for ontology evaluation and selection. In Denny Vrandecic, Mari del Carmen Suarez-Figueroa, Aldo Gangemi, and York Sure, editors, *EON - Evaluation of Ontologies for the Web*, Edinburgh, Scotland, May 2006. WWW2006 Workshop.

[GW00] N. Guarino and C. A. Welty. A Formal Ontology of Properties. In *Knowledge Acquisition, Modeling and Management*, pages 97–112, 2000.

[GW04] N. Guarino and C. A. Welty. An Overview of OntoClean. In S. Staab and R. Studer, editors, *Handbook on Ontologies in Information Systems*, International Handbooks on Information Systems, chapter 8, pages 151–172. Springer, 2004.

[Hea98] M. Hearst. Automated Discovery of WordNet Relations. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[HEHS04] P. Haase, M. Ehrig, A. Hotho, and B. Schnizler. Personalized Information Access in a Bibliographic Peer-to-Peer System. In *Proceedings of the AAAI Workshop on Semantic Web Personalization, 2004*, JUL 2004.

[HMS04] U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^-$ Description Logic to Disjunctive Datalog Programs. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Proceedings of the KR2004*, pages 152–162. AAAI Press, 2004.

[HPSB+04] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. World Wide Web Consortium, May 2004. W3C Member Submission, `http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/`.

[HPSvH03] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, December 2003.

[HSH+05] Jens Hartmann, York Sure, Peter Haase, Raul Palma, and Mari del Carmen Suarez-Figueroa. OMV – Ontology Metadata Vocabulary. In Chris Welty and Aldo Gangemi, editors, *ISWC 2005 - In Ontology Patterns for the Semantic Web*, Galway, Ireland, Nov 2005.

[HV05] Peter Haase and Johanna Völker. Ontology Learning and Reasoning - Dealing with Uncertainty and Inconsistency. In Paulo C. G. da Costa, Kathryn B. Laskey, Kenneth J. Laskey, and Michael Pool, editors, *Proceedings of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*, pages 45–55, NOV 2005.

[KPSH05] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, and James Hendler. Debugging Unsatisfiable Classes in OWL Ontologies. *Journal of Web Semantics, Volume 3 Issue 4*, 2005. (To Appear).

[LSV04] Sik Chun Lam, Derek Sleeman, and Wamberto Vasconcelos. ReTAX+: A Cooperative taxonomy revision tool. In *Proceedings of the AI-2004 Conference*, Cambridge, UK, December 2004.

[MM04] F. Manola and E. Miller. Resource Description Framework (RDF). Primer. W3C Recommendation 10 February 2004, 2004. available at http://www.w3.org/TR/rdf-primer/.

[Mot05] Boris Motik. On the properties of metamodeling in OWL. In *Proc. of the 4th International Semantic Web Conference, Galway, Ireland, November 2005*, NOV 2005.

[MSUW02] Stephen J. Mellor, Kendall Scott, Axel Uhl, and Dirk Weise. Model-Driven Architecture. In *Advances in Object-Oriented Information Systems, OOIS 2002 Workshops, Montpellier, France, September 2, 2002, Proceedings*, volume 2426 of *Lecture Notes in Computer Science*, pages 290–297. Springer, 2002.

[NVCN04] R. Navigli, P. Velardi, A. Cucchiarelli, and F. Neri. Extending and Enriching WordNet with OntoLearn. In *Proc. of the GWC 2004*, pages 279–284, 2004.

[PH02] Jeff Pan and Ian Horrocks. Metamodeling architecture of web ontology languages. In *Isabel Cruz, Stefan Decker, Jérôme Euzenat, and Deborah McGuinness, editors, The Emerging Semantic Web, Frontiers in artificial intelligence and applications*. IOS press, 2002.

[PHS05] Jeff Pan, Ian Horrocks, and Guus Schreiber. OWL FA: A Metamodeling Extension of OWL DL. In *Proc. of the International Workshop on OWL: Experience and Directions (OWL-ED2005)*, 2005.

[Sch05] S. Schlobach. Debugging and Semantic Clarification by Pinpointing. In *European Semantic Web Conference ESWC*, Herakleion, Greece, 2005.

[SM00] Steffen Staab and Alexander Maedche. Axioms are Objects, too - Ontology Engineering beyond the Modeling of Concepts and Relations. Technical Report Internal Report 399, Institute AIFB, Karlsruhe University, 2000.

[VVS05] Johanna Völker, Denny Vrandecic, and York Sure. Automatic Evaluation of Ontologies (AEON). In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *Proceedings of the 4th International Semantic Web Conference (ISWC2005)*, volume 3729 of *LNCS*, pages 716–731. Springer Verlag Berlin-Heidelberg, NOV 2005.